A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

CONF-8306139--1

LA-UR--83-2839

DE84 001033

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE TRACK RECONSTRUCTION OF NORMAL MUON DECAYS IN THE LAMPF TPC: ONE WORKING SCHEME

AUTHOR(S) R. J. McKee

SUBMITTED TO Proceedings of Time Projection Chamber Workshop, TRIUMF, Vancouver, B.C., Canada, June 23-25, 1983.

DISCLAIMER

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO. 836 R4
ST. NO. 2629 5/81

TRACK RECONSTRUCTION OF NORMAL MUON DECAYS IN THE LAMPF TPC: ONE WORKING SCHEME

R. J. McKee
Los Alamos National Laboratory, Los Alamos, NM. 87544

ABSTRACT

A working scheme for track reconstruction of normal muon decays in the LAMPF TPC is here outlined. Muon tracks stopping in the TPC and helical electron tracks from muon decay are both identified and fitted for complete event reconstruction. Because of certain geometrical characteristics of the TPC, novel techniques are deployed to find the tracks. Normal road tracing methods do not work reliably; they are replaced by, among other things, a random search technique that locates the helix's planar projection and a carefully worked-out method for correctly putting each coordinate on its proper turn in the helix.

PROBLEMS FACED BY THE PATTERN RECOGNITION ALGORITHMS

My colleague, Wayne Kinnison, has described elsewhere[1] in some detail what the LAMPF TPC looks like. I shall here briefly review the salient features of the TPC with an emphasis on how those features relate to the special problems any track finding algorithm must face in sorting out track topologies.

The readout plane of the TPC consists of 21 identical rectangular modules. The rectangles are snuggly fitted in a pattern roughly circular in shape. With an effective radius of 50 cm, the circular pattern stretches far enough to fully contain any helical track from muons decaying in the center when the field is raised to 6.7 kilogauss, the nominal field setting. The drift length is 52 cm. Each module is strung with 15 sense wires, each wire possessing a retinue of 17 pads lined up cozily underneath. One centimeter separates neighboring wires from one another and also describes the center-to-center pad separation.

But despite the existence of 315 wires and 5355 pads in the system, dead regions abound. Each module with its G10 frame contributes 39.5% of its total area to the dead regions. These dead regions meander throughout the whole circular pattern of modules, complicating the topology for helical tracks (see Fig. 1). Any given track, for example, might decide to spend most of its time traveling over G10 instead of wires and pads, more than the 39.5% might suggest.

The digitizers, with their flash ADC's and their ability to record the time evolution of everything a pad senses, permit the occurrence of coordinates with the same (x,y) but different z's. Helical tracks of more than one turn, as many turns as can fit in the drift space, are fully digitized. This is a very fine feature for learning as much as possible about each track. But

the algorithms must be prepared to handle the special demands introduced by multiple turning helixes.

Although our interest centers on the helical electron track, we cannot ignore what the incoming muon is doing. Learning about the muon track serves two useful purposes:

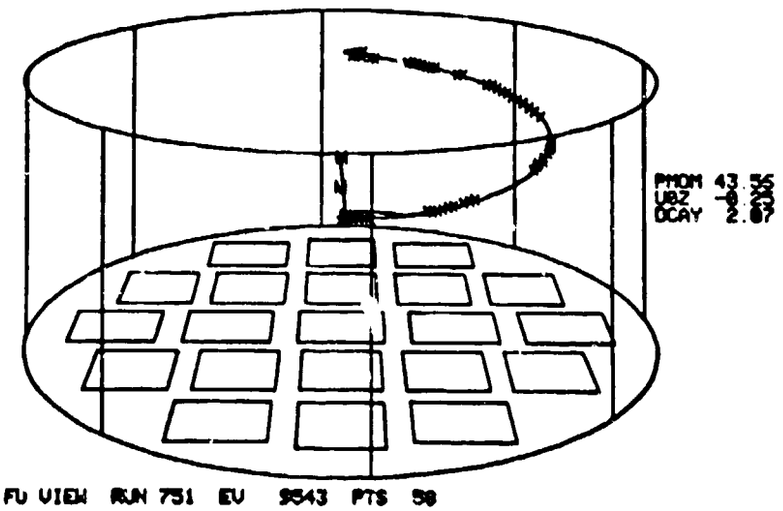
(a) Helps the algorithms get started on the electron track.
(b) Leads to the muon's decay time, whose average should come out to be 2.2 μsec.

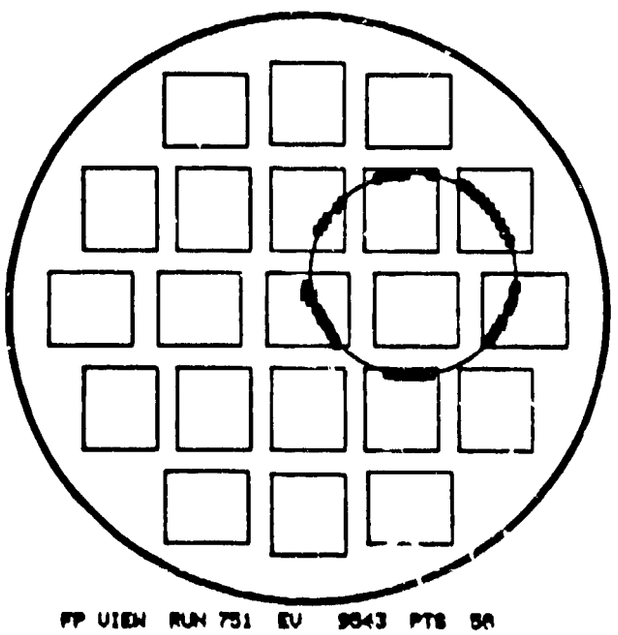
The muon enters the TPC nearly parallel to the magnetic field and, being a degraded surface muon, loafs along so leisurely that it stops in the gas. The muon orbit is not, strictly speaking, a straight line. Multiple scattering and any transverse component to the momentum turn the orbit into a fuzzy decaying helix, especially near the point where the muon stops. But details of the muon's orbit concern us not too much. It is that stopping point that chiefly concerns us.

The muon orbits are prompt, the electron orbits delayed. That is, the trigger counter that senses a muon entering the TPC starts the digitizer clock going, the clock that counts time buckets, and that in turn insures correct measurements of the muon's z coordinates. Not so for the electron. The muon sits there waiting to decay. In the meanwhile the clock is relentlessly ticking off the buckets. It ticks an average of 2.2 μsec before the muon decays, a time which in our TPC is equivalent to a z-distance of 14.3 cm. So the z coordinates of the electron are going to be too big, sometimes by only a little, sometimes by a lot. This delay of the electron, this apparent drifting up of the helix, must be kept firmly in mind by the algorithms.

Unfortunately, there is no sure-fire way of tagging the coordinates as to particle type. Muons, being more heavily ionizing, induce on the average a bigger signal on the sense wires than do the electrons. High level discriminators viewing the central module, where all the muons congregate, take advantage of the greater induced charge as does the software in looking at the digitized signals from the wires. But muon coordinates can sometimes be tagged as electron coordinates and vice versa. Simple geometrical considerations, like knowing that only the central module can see muons, help to some extent. But in the end, only an appeal to topology can sort out who's really who.

In Figs. 1 through 3 I present a typical event fully reconstructed. This example shows the muon track, marked by the M's, and the helical electron track, whose coordinates are indicated by X's or boxed X's. For these computer-eye views, the height of the helix has already been adjusted to accommodate the





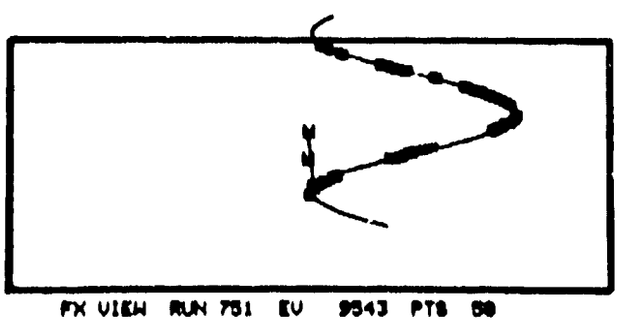


Fig. 1. A reconstructed event in the full TPC. Perspective view, top view, and side view.

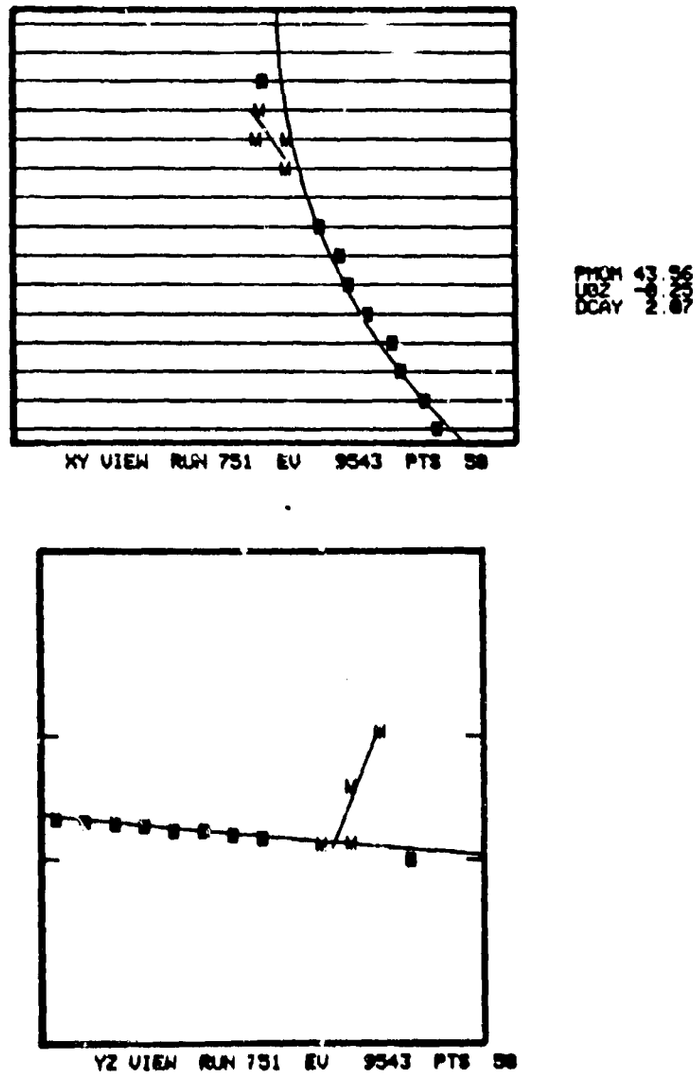


Fig. 2. Top and side views of tracks in central module.

muon decay time. In Fig. 1 I show what the tracks look like in the full TPC. Figure 2 is a close-up glance at the central module, which details the muon track and a piece of the helix. In the top view the computer drew all the wires but for clarity

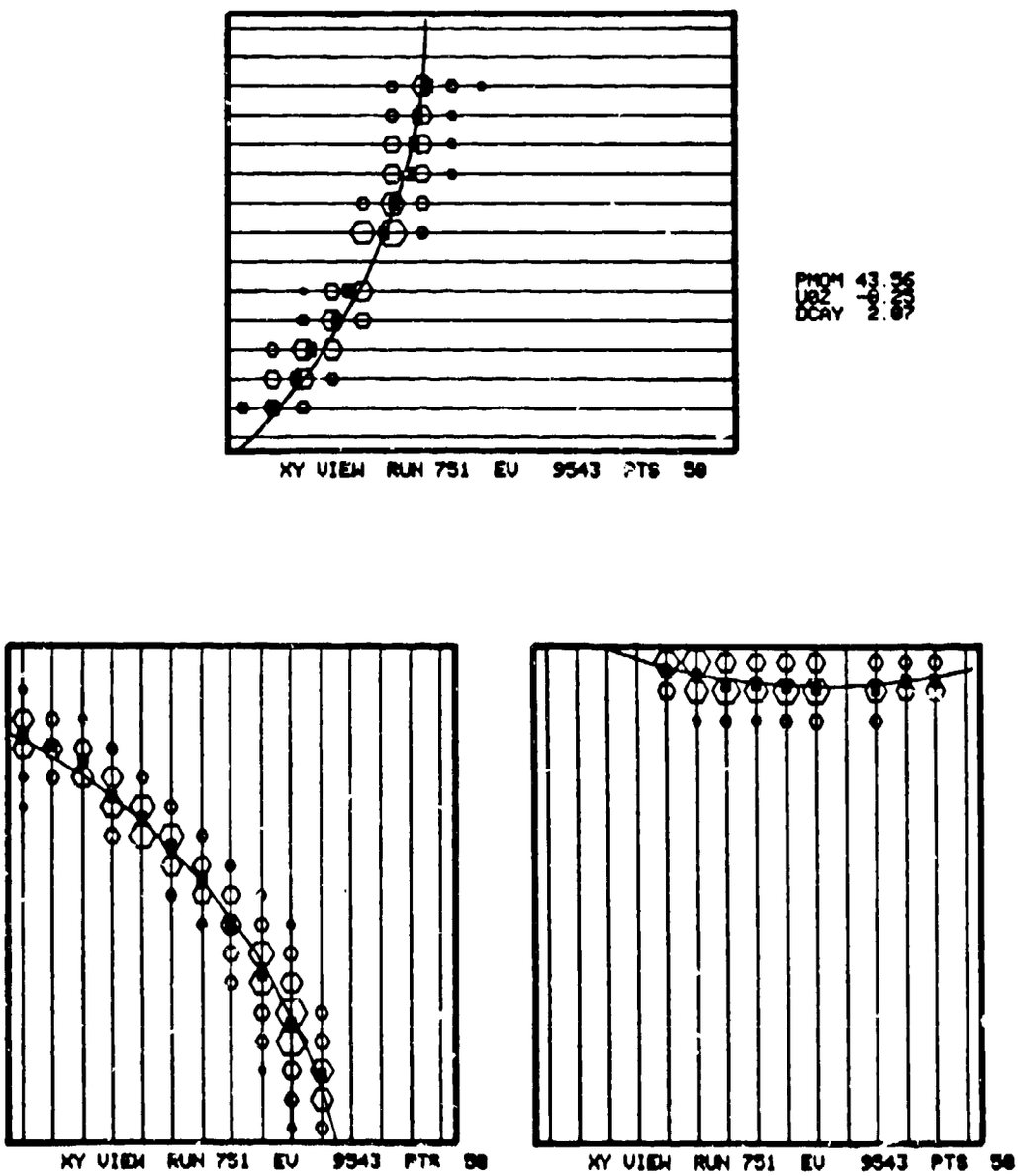


Fig. 3. Top view of helical track in three of the modules.

none of the pads. Figure 3 shows top views of three of the other modules over which the helix sweeps. Again the wires are shown,

but in addition some of the pads, the ones that lit up, are indicated by the hexagons, whose areas are proportional to the induced charge.

The algorithms face other hurdles. Coordinates that should have been present may be missing: In hardware the wire may have failed to fire; in software the algorithm reconstructing coordinates may have rejected the wire turn-on. Or there may be extra coordinates. Multiplexer ambiguities can contribute spurious ones. Others may result from real but unwanted tracks like cosmic rays. Finally, there are imprecisely determined coordinates, coordinates displaced from where they should be. Imprecise coordinates put a somewhat greater strain on the algorithms. Tube sizes about tentative roads through the chamber must be maintained larger than they would otherwise need to be with a higher probability of finding a wrong solution to the topology or missing it altogether.

The following discussion covers four topics: How the muon track is found, how the helix track is found, how the helix is fitted, and finally how muon track and helix are put together to yield the full topology.

MUON TRACK FINDER

The algorithm that searches for the muon track starts off with a pessimistic assumption: Because of possibly wrong tagging, any coordinate found in the fiducial region might turn out to be the last coordinate on the muon orbit. The fiducial region is that volume in the chamber where, according to the trigger, the muon is supposed to stop. The algorithm takes every point in the fiducial volume, regardless of its initial tagging, and tries to construct a muon track on that point, one stopping there. In the end the algorithm winds up with a clutch of muon track candidates, one of which should be the true one. The candidates are ranked in the order in which they are tried in an effort to find the helix.

How is each candidate found? A blunted cone, its flattened nose pointing downward, is drawn about the candidate point lying in the fiducial volume. The blunted nose has a radius of 1.1 cm, enough to touch the two neighboring wires. The cone flares out at the rate of 1 cm for every 5 it gains in elevation. Only points tagged as muons and scooped up by the inverted cone, as well as the candidate point itself at the nose of the cone, are considered for the candidate track. A straight line is fitted to this initial set of points, then the points are iterated in a tube of 1 cm radius; that is, a tube of 1 cm radius is drawn about the straight line fit, points outside the tube are discarded, a new fit is made, a new tube drawn, etc., until no more points are tossed out. If after all that the candidate point has succeeded in getting itself tossed out or if there are

no tagged muon coordinates left, the candidate track is given the heave. It's also chucked if its slope is not steep enough (directional cosine must be ≥ 0.9).

One interesting thing: The candidate track need only consist of a single coordinate, the candidate point that started the search going. In this case the fitted "track" is a vertical line drawn through the point. But from what I said in the last paragraph, the one coordinate must be an already tagged muon.

Now for the order of ranking. Highest honors go to the candidates with the most coordinates. A candidate that is a subset of the true track is therefore ranked lower, an obviously desirable feature. Candidates that tie in number of coordinates are ranked by obliquity: Steeper tracks are given the preference. Candidates that tie in number and are perfectly vertical (several one-coordinate candidates, for example) are ranked by how close they come to the center of the chamber: the closer, the higher the rank.

In the search for the electron helix, the highest ranked muon candidate is tried first. If no helix is found, the next highest ranked muon candidate is tried, etc., until finally we have in hand a totally acceptable solution to the topology. Or until all candidates drop by the wayside. In which case the event is dropped.

HELIX TRACK FINDER

The search for the helix forms the heart of the problem. By comparison, finding the muon orbit is only a minor diversion. Knowing the location of the muon track is helpful in verifying the topology as a whole, but the real test of the algorithms' mettle comes when they turn their attention to bringing the electron helix into the light of day.

The algorithms launch a two-phased attack. The first phase is designed to find the circle, which is the helix's projection onto the xy-plane. Having found the circle and fitted it, the algorithms use the result to unravel ϕ vs. z, where ϕ is the azimuth angle of each coordinate on the circle. At first blush this second phase seems trivial, that there's nothing to unravel. Isn't z a linear function of ϕ and needs only to be fitted with a straight line? But as I will show later, some of the most formidable difficulties of all stand in the way of the solution to this seemingly trivial problem.

But first the circle finder. Here the algorithms deploy strategies which may be unique in the history of track finding. With the existence of circles of any radius and orientation superimposed on our pattern of modules (the only common motif being that the circles pass through the central module), there is no clear way of getting a road started in a systematic fashion, then following it. Coordinates of finite precision and their

closeness to one another in a module, sudden and large jumps across dead areas between modules, are apt to defeat the best efforts at systematic road tracing. The algorithms don't even try. What they do is pick at random quartets of x,y-points, fit a circle to each quartet, concoct a single parameter from the three parameters of the fit, and histogram the result. Many quartets are chosen, 50 to be exact. The reasoning behind this procedure is simple: Because of the possible existence of spurious coordinates, some quartets may give wrong results. At least the algorithms don't assume that a given quartet will reveal anything reasonable. But most quartets do; most quartets, being a subset of the real circle, yield parameters that cluster about some central value. So the problem of finding the circle is metamorphosed into the simpler problem of searching for a peak in a histogram.

But don't think for one moment that the algorithms accept every quartet willy-nilly. Each fitted circle must pass certain tests: The rms cannot be too large, the radius as well as the center of the circle must be within bounds, the circle has to pass close to the place where the muon stopped (or rather, the stopping point of the muon track candidate under consideration). This last test is a rather important one, for if there happen to be two helixes in the chamber, the decay electron and some unrelated spurious track, this test greatly increases the chances of the algorithms' grabbing onto the right track.

In binning the parameter found from fitting each quartet, the algorithms take some care in selecting the bin size and number of bins. It is desirable to have the peak sticking up prominently above background and at the same time have a way of finding its position without doing any fancy fitting. The algorithms choose the number of bins to be just equal to the number of accepted quartets. That makes the average count per bin be one. If in addition the range of the histogram is chosen to just barely contain the lowest and highest of the parameters, the bad quartets will scatter themselves more or less uniformly throughout the whole histogram, with much less than one count per bin on the average, while the good quartets will concentrate in a small area, perhaps in one or two bins, each bin having many counts. That makes the chore of finding the peak an easy one.

The algorithms, in forming the initial set of points for the circle, draw on the coordinates from the good quartets. The final set is extracted by iteration in a circular band of a given width. The set is fitted to a circle, a band of a certain width is laid around the circle, points not belonging to the candidate muon track but falling in the band form the next set, this set is fitted to a circle, etc., etc. And so the iterations go until the algorithms find the same set of points two times in succession or until the set falls below a membership of six points, in which case the search for the circle is declared a bum

steer. The half width of the circular band is larger on the first round, then squeezed tighter once for all subsequent rounds.

With the circle found, the problem is now ϕ vs. z. For that I will write down the helix orbit equations we habitually use:

$$x = a - \rho\cos(\kappa s - \phi_o) \tag{1}$$

$$y = b + \rho\sin(\kappa s - \phi_o) \tag{2}$$

$$z = z_o + u_{oz}s \tag{3}$$

$$\kappa = \frac{0.29979B_o}{p} \tag{4}$$

$$\rho = \kappa^{-1}\sqrt{1 - u_{oz}^2} \tag{5}$$

The first three equations describe the helical orbit parametrically in terms of the path length s. The three parameters a, b, and ρ refer to the circle and are the location of the circle's center and its radius. κ, in units of cm^{-1}, is related to the helix pitch, the distance between turns:

$$\text{pitch} = \frac{2\pi|u_{oz}|}{\kappa} \tag{6}$$

The parameter ϕ_o is the initial azimuth angle, related to the initial (s = 0) directional cosines in x and y:

$$u_{ox} = -\rho\kappa\sin\phi_o \tag{7}$$

$$u_{oy} = \rho\kappa\cos\phi_o \tag{8}$$

The third directional cosine is u_{oz}. How it connects to ρ and κ is shown in Eq. (5), a result that flows from the requirement that the sum of squares of all three directional cosines must add up to 1.

For completeness, I have included Eq. (4), which shows how κ relates to the magnetic field B_o (in kilogauss) and the momentum p (in MeV/c).

I shall now recast the helix equations into a different form by eliminating the path length parameter s:

$$z = Z + \lambda\phi \tag{9}$$

$$\phi = \tan^{-1}\frac{y - b}{a - x} \tag{10}$$

$$\lambda = \frac{u_{oz}}{\kappa} = \pm\frac{\text{pitch}}{2\pi} \tag{11}$$

$$Z = z_o + \lambda\phi_o \tag{12}$$

Equation (9) is it, the ϕ vs. z connection alluded to earler. With the circle already known (a, b, and ρ extracted from a circle fit), the azimuth angle ϕ can be found for every (x,y) coordinate used in f tting the circle, then a linear fit to z performed, the process yielding the two parameters λ and Z. With λ found and ρ already known, the parameters κ and u_{oz} can be extracted through the auspices of Eqs. (5) and (11).

But here is the rub: Equation (10), the prescription for generating the azimuth angles, is incomplete. The arctangent at best creates angles in the range $[-\pi,\pi]$, and even that assumes the use of the two-argument arctangent function of FORTRAN fame. What is missing in Eq. (10) is the $2\pi n$ term, where n is the sheet number of the branch of the helix to which each coordinate belongs. But a priori, the algorithms do not know the sheet numbers and there is no easy way of assigning them. Even a short track making only a fraction of a turn might cross the branch cut, jumping from one sheet to the next suddenly. A plot of Eq. (9) is not a straight line then but a series of disjointed straight lines looking like the profile of a saw blade. That may not seem so bad at first glance, but you must remember that the algorithms don't see a neat even saw blade, continuous except at the jumps, but rather they see discrete points of varying degrees of precision perching at varying intervals along the saw blade.

Thanks to the dead regions between modules, the azimuth separation between adjacent points can be very large, with a change of sheet number quite likely.

And there is an additional complication. Just because a coordinate seems to fit very neatly on the circle, it doesn't follow that the point belongs to the helix, as that marvelous pattern recognizer, the human eye, might tell you after a quick glance at the side view. So the algorithm assigned to grapple with the sheet numbers must not be so naive as to assume that every coordinate found on the circle really belongs.

With the points on the circle arranged in the order of increasing z, the basic strategy of the algorithm is to feel its way from point to point slowly and carefully, pausing each time to decide first whether the point really belongs and if so whether it makes a jump in sheet number. The algorithm fulfills its task by performing a running fit to Eq. (9) using the points already felt, those points having had their sheet numbers determined. Using the fit, the algorithm calculates z for the next point, with and without a change in sheet number, and compares it to the real z. Three projected z's are considered, one each for sheet changes of 0, -1, and +1. If no projected z satisfies the tolerance, currently set at 2.5 cm, the algorithm chucks the point. Otherwise the point is retained with a sheet change reflecting the best comparison between real and projected z.

But getting started -- how is that done? It takes two points to start a road in ϕ vs. z. Bear in mind that however the two are chosen, one or both points might not belong to the helix and if they do, they might not dwell on the same sheet. So the road-tracing algorithm must expect to stumble into false starts occasionally and be prepared to try again.

The algorithm starts by choosing the first point at random from among the ordered points on the circle. The second point is chosen to be the next one in order. The advantage of trying two neighboring points is that both are likely to reside on the same sheet. At any rate, the algorithm starts off assuming the same sheet number. If that proves wrong (the algorithm soon learns by losing the road), adjacent sheet numbers can then be tried. If after that the road still wanders into limbo, the algorithm shrugs and mutters to itself, "Bad starting pair. Probably the first point or its neighbor doesn't belong. Better start somewhere else." And so it does, again with the first point chosen at random.

FITTING THE HELIX

As can be gleaned from the previous section, track fitting forms an inseparable part of track finding. The circle is found and fitted before a single z coordinate is scrutinized. Then, as

the algorithm hacks its way through the saw-toothed grass of ϕ vs. z in its search for sheet numbers, it fits z to ϕ at every opportunity. But nevertheless, many characteristics of the fitting will be more brightly illuminated if they are pulled out of their setting and given a separate look.

First, be it noted that there is no unique or even obvious way of fitting a helix. I suppose the most "obvious" way would be to find the shortest distance between each x,y,z-data point and the helix, treat each minimum distance like a residual, and adjust the helix parameters to minimize the sum of the squares of the residuals, with weights or without. But the best method in my opinion, because it is the simplest, is the 3-and-2 fit: Do the x,y-view first, a three-parameter fit to the circle, then follow it with ϕ vs. z, a two-parameter fit to get the rest of the description of the helix. Although closely associated with the track finding algorithms, the 3-and-2 method owes nothing to them. The method can be appreciated on its own merits.

First, the circle fit. Uniqueness does not seem to be a virtue here either. One scheme, however, has greater intuitive appeal over all the others. I shall lead up to it by first considering a more general fitting scheme.

Fitting a circle described by a, b, and ρ to a bunch of data points (x,y) involves a least squares sum of some sort of residuals (I am dealing with only unweighted fits). Here are the residuals of interest:

$$\delta r_n = \rho^n - r^n \tag{13}$$

$$r = \sqrt{(x - a)^2 + (y - b)^2} \tag{14}$$

where n is any fixed positive number, not necessarily an integer.

Intuitively, the choice of $n = 1$ seems best. δr_1 is the uncomplicated, straight-forward radial residual, the common kind of thing that shows up in most least squares fitting. Unfortunately, unlike fitting a straight line, the least squares sum of δr_1 leads to non-linear equations in the parameters a, b, and ρ, a not very pretty feature.

However, for $n = 2$ (and only $n = 2$), linear equations do result. Or to be more precise, the equations are linear in a, b, and the combination $\rho^2 - a^2 - b^2$.

But while $n = 2$ produces cleaner mathematics, its use is not as intuitively persuasive. So here is what the algorithms do: Throughout most of the work the $n = 2$ residuals are used. It is only when we want final values for the circle parameters that the algorithms switch to the $n = 1$ residuals. And the non-linear

mathematics? Pretty easy at this stage. The mathematics of n = 2 have given good starting estimates for quickly iterating to the final values via Newton's method (Eq. (19) below).

At this point I shall write down the n = 1 mathematics. I do this not only for completeness but also to show how the mathematics of n=1 ultimately lead to a calculation of momentum resolution.

$$r_i^2 = (x_i - a)^2 + (y_i - b)^2$$

$$h_1 = \sum 1$$

$$h_2 = \sum r_i^{-1}(x_i - a)$$

$$h_3 = \sum r_i^{-1}(y_i - b)$$

$$h_4 = \sum r_i^{-1}[r_i^{-2}\rho(x_i - a)^2 - \delta r_i^i]$$

$$h_5 = \sum r_i^{-3}\rho(x_i - a)(y_i - b)$$

$$h_6 = \sum r_i^{-1}[r_i^{-2}\rho(y_i - b)^2 - \delta r_i^i] \qquad (15)$$

$$t_1 = \sum \delta r_i^i$$

$$t_2 = \sum r_i^{-1}(x_i - a)\delta r_i^i$$

$$t_3 = \sum r_i^{-1}(y_i - b)\delta r_i^i \qquad (16)$$

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_2 & h_4 & h_5 \\ h_3 & h_5 & h_6 \end{pmatrix} \qquad (17)$$

$$E = H^{-1} = \begin{pmatrix} e_1 & e_2 & e_3 \\ e_2 & e_4 & e_5 \\ e_3 & e_5 & e_6 \end{pmatrix} \qquad (18)$$

$$\begin{pmatrix} \rho \\ a \\ b \end{pmatrix} + \begin{pmatrix} \rho \\ a \\ b \end{pmatrix} - E\begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \qquad (19)$$

$$E_\rho = e_1 \tag{20}$$

$$\sigma_\rho = \sigma_R E_\rho^{1/2} \tag{21}$$

The last equation shows how σ_R, the sigma of the n = 1 residual distribution, which is approximately Gaussian, relates to the sigma of the error distribution for ρ, the radius. The connection, E_ρ, is the error matrix element. I call the latter the geometrical factor. The quantity σ_R, which depends on the details of coordinate reconstruction, is treated in this analysis as a universal constant. All helixes have it. Not so E_ρ. That quantity depends on the geometry of the helix under study. Some helixes have a large E_ρ, others a much smaller one. And so σ_ρ also reflects the geometry: σ_ρ describes the (almost) Gaussian way the error in ρ distributes itself under a specific set of geometrical considerations. Charge the geometry and the width of the distribution changes accordingly.

With the results of the circle fit in hand, the two-parameter fit of ϕ vs. z can be performed. The algorithm chooses to perform the fit by invoking the least squares sum of the z residuals, a straight-forward procedure once the sheet numbers are assigned. I now write down a series of equations paralleling Eqs. (15) through (21), showing the mathematics of the fit:

$$\begin{aligned} h_1 &= \sum 1 \\ h_2 &= \sum \phi_i \\ h_3 &= \sum \phi_i^2 \end{aligned} \tag{22}$$

$$\begin{aligned} t_1 &= \sum z_i \\ t_2 &= \sum \phi_i z_i \end{aligned} \tag{23}$$

$$H = \begin{pmatrix} h_1 & h_2 \\ h_2 & h_3 \end{pmatrix} \tag{24}$$

$$E = H^{-1} = \begin{pmatrix} e_1 & e_2 \\ e_2 & e_3 \end{pmatrix} \tag{25}$$

$$\begin{pmatrix} Z \\ \lambda \end{pmatrix} = E \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \tag{26}$$

$$E_\lambda = e_3 \tag{27}$$

$$\sigma_\lambda = \sigma_Z E_\lambda^{1/2} \tag{28}$$

Like Eq. (21), the last equation shows how σ_Z, the sigma of the z residual distribution, also nearly Gaussian, connects to the sigma of the error distribution in λ, the helix pitch parameter. The connection is again an error matrix element, another geometrical factor. For like σ_R, σ_Z is essentially a universal constant while E_λ varies from helix to helix, just as E_ρ does.

I am now in a position to write down the complete error matrix results for the momentum and angle resolutions. For that purpose I will let μ stand for u_{oz}, the cosine of the electron's angle of emission. Through the auspices of Eqs. (21) and (28), I can write down the sigmas for ρ and μ:

$$\sigma_p = p\kappa^2\sqrt{\rho^2 E_\rho \sigma_R^2 + \lambda^2 E_\lambda \sigma_Z^2} \tag{29}$$

$$\sigma_\mu = \rho\kappa^3\sqrt{\lambda^2 E_\rho \sigma_R^2 + \rho^2 E_\lambda \sigma_Z^2} \tag{30}$$

This then completes the fitting of the helix.

PUTTING TOGETHER THE COMPLETE TOPOLOGY

Finding the helix does not automatically end the job. The helical track must be reconciled with the muon track candidate. Already some reconciation exists since in the search for the helix the algorithms have forced a certain coziness between the x,y-projection of the helix and the muon track's end point. But that is not enough. The end point might be wrong or the helix ill-found in some way. More tests need to be made. And there are some rather easy ones waiting in the wings. These tests are based on a very simple observation:

The coordinates fitted to the helix must in the end all lie either above or below the muon's end point and none must stick out of the TPC.

Why do I say "in the end"? Because muons are prompt, electrons delayed. At the time the helix is first reconstructed, it is in the wrong place. It is up too high, a distance equivalent to the time the muon sat around waiting to decay. To test the topology, we must move the helix down to its rightful place. Or rather, the very act of trying to move it to the right place constitutes the test.

How is the "right" place determined? By moving the helix until it comes closest to the end point of the muon track.

That bare statement needs amplification. Naturally, it is perfectly correct to say that the fitted helix should come as close as possible to the end point since the electron track originates there. Only finite precision and the muon's finite decay time sunder electron from muon. But the helix, looked upon as a mathematical construct, has an infinite number of closest points, all bearing the same x and y but differing in z by one or more pitches. Moving the helix down until the first of these points becomes closest may not be enough; the muon may have sat around longer than that, long enough to allow the helix's apparent upward drift to encompass additional pitches.

So the algorithm must depress the helix that first fraction of a pitch plus any additional integer pitches until all the helix coordinates reach one side of the muon's end point or, if they were all already on the up side, until one more pitch would split them up. The algorithm is not a complete stickler for this, however. Because of finite precision in the coordinates, the helix might be left with one or two points a little over the border marked by the muon's end point or a little out of the TPC's end planes. Under certain conditions, it is even possible that the helix might find itself raised a little. This can happen when the muon's end point is already quite close to a helix loop, the end point being a little above by a small fraction of a pitch, and other conditions forbid a downward movement of the helix.

The known drift velocity and the distance the helix has to be depressed disclose how long the muon lived. A histogram of that over many events should look like an exponential with a decay time of 2.2 μsec, which, if it does, serves as a quick and simple overall test of how well the algorithms are doing their jobs. In Fig. 4 I present such a histogram. The average decay time is indeed quite close to 2.2 μsec.

Of all the problems the algorithms run into, the biggest failing resides in the matching of the muon track to the helix. The reasons for that are pretty clear: the paucity of muon coordinates in a typical event, muon coordinates wrongly identified, the stringent requirements for matching. The

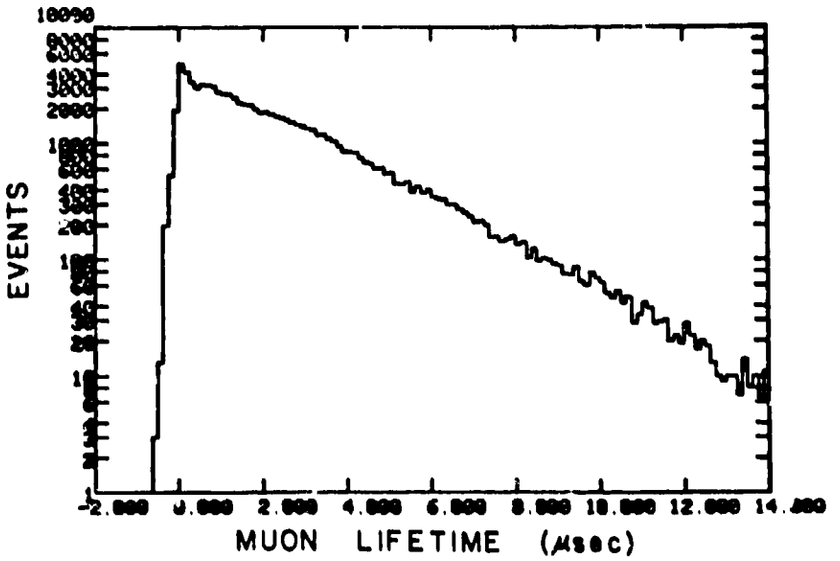


Fig. 4. Muon decay time.

algorithms purposely make matters tough based on the philosophy that it is better to miss the solution entirely than to grab onto a wrong one. Of course it can be argued that, as far as introducing biases into the Michel spectrum goes, missed solutions are surely as bad as wrong ones. But the likeliest reason for missing the boat in the case of a failure to match muon to electron centers on the muon. An event with a wrongly found muon track smells very similar to a trigger inefficiency as far as the event's final disposition is concerned. And a muon that fails to trigger the TPC has no bias associated with it; the reason for failing to trigger is probably unrelated to the way the muon emits the decay electron.

With the successful re-positioning of the helix, the analysis of the topology is complete.

PERFORMANCE, RESOLUTION, AND ACCEPTANCE

To ask the question, How well do the algorithms perform? would have little relevance without some mention of the TPC's resolution and acceptance. Certain topologies, like electrons emitted very far forward or backward, generate too few coordinates for a helix fit and hence cannot be done. Other

topologies turn out to be only marginally doable; such a catagory includes some types of flat tracks -- marginal because of multiplexer ambiguities. Still others can be done reliably but produce imprecise measurements of the momentum; helixes with few coordinates or a small turning angle are that type.

So the question should read: How well do the algorithms perform in a cut domain, those regions of the TPC blessed with doable topologies and good resolution? The answer is: Quite well. Indeed, when you consider that our understanding of the TPC is still in the preliminary stages and that we have yet to complete the instrument's calibration, the algorithms do remarkably well. Figure 5, which shows the Michel spectrum, bears out that assessment. In looking at the figure, two salient features leap to the eye. One is how closely the solid line noses after the data. That curve was calculated from the V-A theory of muon decay with resolution and acceptcance folded in (radiative corrections are not included, however). The agreement between calculated curve and data demonstrates that, despite our currently limited understanding of the TPC, we can handle the cuts pretty well. The second striking feature is the sharpness of the Michel edge. That sharpness speaks unambiguously of the

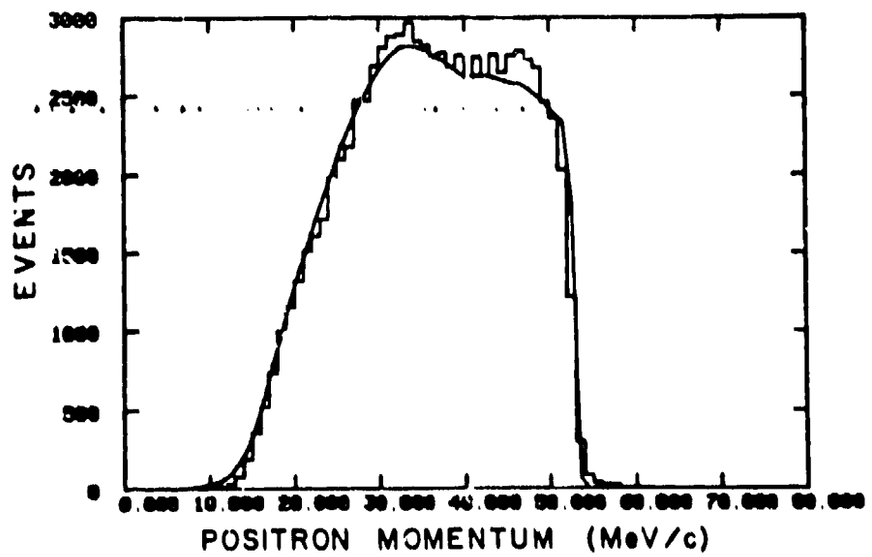


Fig. 5. The Michel spectrum over the cut TPC.
Solid line is V-A theory with acceptance and resolution.

quality of the momentum resolution we are currently seeing in the TPC. A direct measurement of the momentum resolution over the observed Michel spectrum is shown in Fig. 6. The quantity $\delta p/p$ is calculated by the error matrix method described in the last section. The resolution averages to a value of 0.7%.

As our understanding of the TPC deepens, so will matters like resolution and acceptance improve. At the present time we are far from achieving the precision in pad interpolation we would like to see. Corrections like the E cross B effect have yet to be studied and applied. That we see already as good a momentum resolution as we do speaks eloquently of the TPC's inherent capability of measuring momenta well. That fine capability automatically springs from built-in geometrical factors: many samplings of the orbit spread over one or more helical turns.

The acceptance we are achieving at the present time, an acceptance based on one magnetic field setting, the field that keeps electrons of 52.8 MeV inside the chamber -- that acceptance is 1/3. The acceptance should also improve as we come to understand the TPC better, although the acceptance will never reach 100%. Yet, through the use of different magnetic fields,

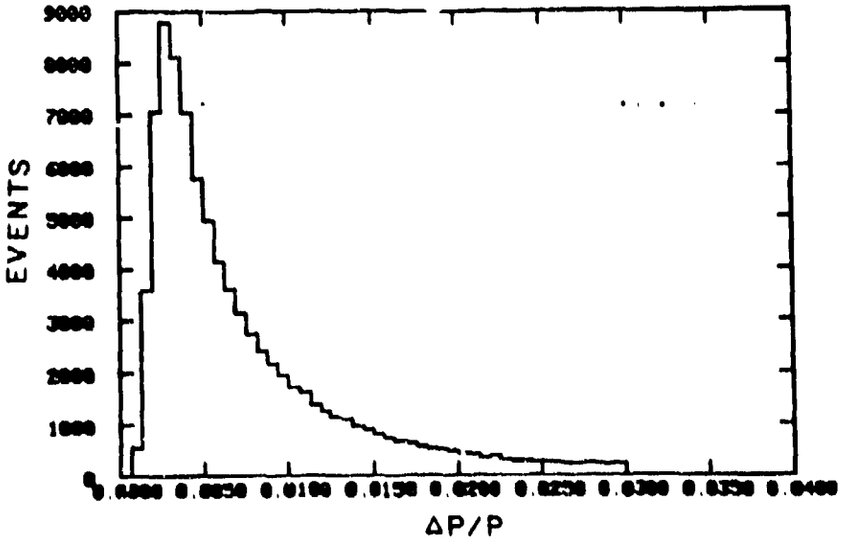


Fig. 6. Momentum resolution ($\delta p/p$) for observed Michel spectrum.

topologies now hard or impossible to get should become doable with good resolution. It may come to pass that we will be able to overlap acceptance ranges through the use of different field settings and in effect achieve an acceptance approaching 100%.

One final question could be asked: How fast are the algorithms? On the VAX it takes about one second of CPU time to analyze an event. Now, that includes everything, the track-finding algorithms as well as things like the algorithm that constructs coordinates from the raw digitizer data. So all I can say is that the algorithms discussed here take less than a second on the VAX (on the average) to reconstruct a topology.

CONCLUSIONS

Within the domain defined by doable topologies and good momentum resolution, the track finding algorithms for our TPC, as things now stand, do quite well. The biggest failing resides in the marching of helix to muon, and we think that such failings, in that they throw out events rather than reconstruct them wrongly, act like a trigger inefficiency as far as introducing biases are concerned. Many rather novel features are employed to help unravel the topologies, including a random number generator for searching for the helical track's planar projection. Also employed are a shotgun approach for locating the muon track and, in the case of the helical side view, a carefully thought-out procedure for extracting the correct branch of the helix to which each coordinate belongs.

One final comment. The emphasis here is on the fact that we have developed a working scheme of track reconstruction, not a final one. Much work has to be done in understanding and calibrating the TPC. By extension, the preliminary status of matters reflects on the algorithms. As our understanding deepens, so too will the algorithms change to reflect that understanding. But in the round, the basic ideas presented here will not, I suspect, change.

1. W. W. Kinnison, "A TPC Spectrometer for Measuring the e^+ Spectrum in μ Decay" (Proceedings, this conference).